MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 3, 2017/2018

DCS5038 – PROGRAM DESIGN

(DIT Groups)

4 JUNE 2018
2.30 p.m. – 4.30 p.m.
(2 Hours)

INSTRUCTIONS TO STUDENT:

1. This question paper consists of 10 pages with 5 questions.
2. **SECTION A**: Answer **ALL** questions.
3. **SECTION B**: Answer **ONLY ONE (1)** question.
4. Please write all your answers in the Answer Booklet provided.

SECTION A (75 Marks)

Instruction: Answer ALL questions from this section and write your answers in the answer booklet provided.

QUESTION 1 (25 Marks)

a. Based on the description given below, draw the **flowchart** to calculate points achieved when user jog at a certain distance. [13 marks]

- Initialize all the required variables.
- Get the input jog *distance* from the user.
- Based on the table given, set the *points* achieved based on the *distance*.

Jog Distance (KM)	Points Achieved
More than or equal to 10	1000
More than or equal to 5	500
More than or equal to 3	200
Others	0

- Display the jog *distance* and *points*.
- Repeat the process by asking the user to continue or to terminate the program. Use a *do while* loop.

b. Based on the description given in (a), write the **pseudocode**.

[12 marks]

Continued...

QUESTION 2 (25 Marks)

a. Convert the following mathematical formulas to the proper C expressions. Use appropriate built-in functions from the *math.h* header file. [4 marks]

i. $K = \sqrt{CD} + \sqrt{A^2}$ ii. $K = \frac{\sqrt{4AC}}{\sqrt{2B^2}}$

b. Write the *nested if-else* statement that determines the *winner* of Rock-Paper-Scissors Game depending on the choice made by *Player 1* and *Player 2* as indicated in the table below. [Hint: Use string compare and string copy function] [9 marks]

Rock-Paper-Scissors Game		
Player 1	**Player 2**	**Winner**
ROCK	ROCK	TIE
	PAPER	PLAYER 2
	SCISSORS	PLAYER 1
PAPER	ROCK	PLAYER 1
	PAPER	TIE
	SCISSORS	PLAYER 2
SCISSORS	ROCK	PLAYER 2
	PAPER	PLAYER 1
	SCISSORS	TIE

c. Based on a research made by a Japanese professor, write a full program that determines the *traits* of a person based on *blood type*. The program should respond to the input of an integer. [12 marks]

In function *main()*
- Get the *blood type* from user.
- Call function *personalTraits(...)* and pass the *blood type* as the parameter.
- Display the *traits*.

In function *personalTraits(...)*
- Declare a prototype for this function.
- Use *switch-case* statement to set the *traits* based on the *blood type* (refer to the table below).
- Return the *traits*.

Continued...

Blood Type Personality Traits	
Blood Type	**Traits**
1	Kind and compassionate individuals who put others' interests and needs before their own.
2	Outgoing and friendly, this type is people person.
3	Strong and rational people
4	Responsible, practical and rule-conscious and organized
Others	Invalid blood type

SAMPLE OUTPUT 1

```
Please enter your blood type (in numbers) [A=1|B=2|AB=3|O=4]:2

This type of blood is describe as outgoing and friendly, this type
is people person.
```

SAMPLE OUTPUT 2

```
Please enter your blood type (in numbers) [A=1|B=2|AB=3|O=4]:5

This type of blood is describe as invalid blood type.
```

Continued...

QUESTION 3 (25 Marks)

a. Trace the output for the following program. [3 marks]

```
#include<stdio.h>

void main()
{
     int arrayNum[] = {5, 10, 4, 25, 4, 12, 15, 20, 11, 30};

     int a=3, b=6, c=9;
     int *x, *y, *z;

     x=&a;
     y=&arrayNum[4];
     z=&b;

     printf("%d %d %d\n", a,b,c);
     printf("%d %d %d\n", *x,*y,*z);
     x=&arrayNum[6];
     printf("%d\n", *x+*y);
     printf("%d\n", *(x+2)**z);
     z=&c;
     printf("%d\n", (*y+1)-4);
     printf("%d %d %d\n", *x,*y,*z);
}
```

b. Write a full program to determine the rank of Malaysian Air Force team based on the following instructions. [12 marks]

Declare a structure of Malaysian Air Force record called *AirForceEmp* that has:

- Air Force Name : *name (string)*
- Air Force Age : *age(integer)*
- Salary : *salary(float)*
- Rank : *rank(string)*

In function *main()* :

- Create a structure variable array called *details* with 3 elements.
- Using *for loop*, ask user to enter *name, age,* and *salary.*
  - Use *if..else* statement to identify the *rank* based on the *salary.*

Salary (RM)	Rank
More than 10,000	Marshal
More than 7000 and less than or equal to 1000	General
More than 5000 and less than or equal to 7000	Colonel
Less than or equal to 5000	Lieutenant

- Using another *for loop* to display the *name, age,* and *rank.*

Continued...

SAMPLE OUTPUT

```
Enter Name      : Ping Chen
Enter Age       : 44
Enter salary    : RM7895
------------------------------
Enter Name      : Muhammad Ali
Enter Age       : 34
Enter salary    : RM5462
------------------------------
Enter Name      : Harjit Singh
Enter Age       : 56
Enter salary    : RM12983
------------------------------

MALAYSIA AIR FORCE TEAM 1 DETAILS
-----------------------------------
Air Force 1
Name    :Ping Chen
Age     :44
Rank    :General
-------------------------
Air Force 2
Name    :Muhammad Ali
Age     :34
Rank    :Colonel
-------------------------
Air Force 3
Name    :Harjit Singh
Age     :56
Rank    :Marshal
-------------------------
```

c. Write a full program that calculates the total price based on the following instructions. [10 marks]

In function *main()* :

- Create files variables named *freadRental* and *fupdateRental.*
- Open the file *movieRental_file.txt* for reading using *freadRental* and *fupdateRental* for append the file.
- Using a *while loop* statement to loop till the end of the file:
  - Read all the data from the file and store them in appropriate variables.
  - Display all the data in the command prompt.
  - Calculate the *total pay* and append the *total pay* into the file.
  - Close the files. The contents of the file and the output screen are shown below.

Content of *movieRental_file.txt* before execution

```
< custName> <movTitle> <price> <dayRent>
Sarah          Logan       2.50      3
```

Content of *movieRental_file.txt* after execution

```
< custName> <movTitle> <price> <dayRent><totalpay>
Sarah          Logan       2.50      3       7.50
```

Continued...

SAMPLE OUTPUT

```
Customer Name          : Sarah
Movie Title            : Logan
Rental Price/day       : RM2.50
Days rent              : 3 day(s)
Total Price            : RM7.50
```

Continued...

SECTION B (25 Marks)

Instruction: *Choose and answer* ***ONLY ONE (1)*** *question from this section and write your answers in the answer booklet provided.*

QUESTION 1 (25 Marks)

Write a full program based on the following instructions. This program prompts the user to enter the item that the user wants to buy from a bakery store and displays the order details.

Declare a *structure* of bakery store record called *Bakery* that has:

- Item Name : *itemName (string)*
- Quantity : *qty (int)*
- Price : *price (float)*
- Total Price : *totalPrice(float)*

In function main() :

- Ask the user to enter *number* of item.
- Create a structure variable array *order* which number of element depends on the *number* of item that user key in.
- Using a *for loop* :
  - Prompt the user to enter *itemName* and *qty*.
- Call function *getPrice(...)* and pass *itemName* as parameter to determine the *price* per unit.
- Call function *getTotalPrice(...)* and pass *price* and *qty* as parameters to determine the *total price* of each item.
- Using a *for loop* :
  - Display all the details of the ordered items as shown in the sample output.
  - Calculate the *total amount* user has to pay.
- Display the *total amount*.

In function *getPrice(...)*:

- Declare a prototype for this function.
- Using *if else* statement, assign the *price* based on the following table. [Hint: Use a string compare]

Item Name	Price (RM)
Pies	4.00
Bread	2.00
Muffin	1.50
Cookies	6.50
Rolls	3.00
Others	0.00

- Return the *price*.

Continued...

In function *getTotalPrice(...)*:

- Declare a prototype for this function.
- Calculate the *totalPrice* based on the formula given below:
  Total Price = Price x Quantity
- Return the *Total Price*.

SAMPLE OUTPUT 1

```
How many items do you wish to order? [1-5]: 3

Enter item 1: Bread
Enter quantity: 3

Enter item 2: Muffin
Enter quantity: 2

Enter item 3: Cookies
Enter quantity: 4


============================================================
                        ORDER DETAILS
============================================================
Item Name       Quantity        Price per Unit  Total Price
Bread           3               RM2.00          RM6.00
Muffin          2               RM1.50          RM3.00
Cookies         4               RM6.50          RM26.00

Total Amount Pay: RM35.00
```

SAMPLE OUTPUT 2

```
How many items do you wish to order? [1-5]: 2

Enter item 1: Pies
Enter quantity: 5

Enter item 2: Rolls
Enter quantity: 1


============================================================
                        ORDER DETAILS
============================================================
Item Name       Quantity        Price per Unit  Total Price
Pies            5               RM4.00          RM20.00
Rolls           1               RM3.00          RM3.00

Total Amount Pay: RM23.00
```

Continued...

QUESTION 2 (25 Marks)

Write a full program that determines the average percentage and survey status of a survey conducted by a newly opened cafe.

Given is a text file named *surveyIn_file.txt* that contains data of invoice id, index of Quality of Service (QoS), Quality of Product (QoP) and Café Environment (CE).
Note: Each category has been rated from 1 (worst) to 5 (best).

Content of *surveyIn_file.txt*

<invoiceID>	*<qos>*	*<qop>*	*<ce>*
1112A	3	3	5
1113B	2	4	3
1114C	5	5	5
1115D	2	2	1
1116E	5	1	1

Another text file named *surveyOut_file.txt* is an empty file.

In function *main()* :

- Declare all necessary variables.
- Create files name: *fread* and *fwrite*.
- Open the file *surveyIn_file.txt* for reading.
- Open the file *surveyOut_file.txt* for writing.
- If the file *surveyIn_file.txt* doesn't exist, display "*File cannot be accessed! Please check.*", else use a *while loop* to read each record until the end of the file:
  - Read the *invoiceID*, *qos*, *qop* and *ce* from the file.
  - Call function *getAvgSatisfaction(...)* by passing the *qos*, *qop* and *ce* as parameters.
  - Call function *getStatus(...)* by passing the *avg* as parameter.
  - Write the *invoiceID*, *qos*, *qop*, *ce*, *avg* and *status* into file *surveyOut_file.txt* as shown below.

Content of *surveyOut_file.txt* after execution

<invoiceID>	*<qos>*	*<qop>*	*<ce>*	*<avg>*	*<status>*
1112A	3	3	5	73.3	Good Job
1113B	2	4	3	60.0	Please Review
1114C	5	5	5	100.0	Good Job
1115D	2	2	1	33.3	Please Review
1116E	5	1	1	46.7	Please Review

  - Display all data in the command prompt as shown in the sample output.

SAMPLE OUTPUT

InvoiceID	QoS	QoP	CE	AVG	Status
1112A	3	3	5	73.3%	Good Job
1113B	2	4	3	60.0%	Please Review
1114C	5	5	5	100.0%	Good Job
1115D	2	2	1	33.3%	Please Review
1116E	5	1	1	46.7%	Please Review

Continued...

- Close the files.

In function *getAvgSatisfaction (...)*:

- Declare a prototype for this function.
- Determine the *avg* based on the following formula.

  Average Satisfaction = ((Qos+QoP+CE)/15x100)
- Return the *avg* value.

In function *getStatus(...)*:

- Declare a prototype for this function.
- Determine the *status* based on the following table.

Average Satisfaction Value	Status
Less than or equal to 60.00	Please Review
More than 60.00	Good Job

- Return the *status*.

End of Page.